

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky

BAKALÁŘSKÁ PRÁCE

2011

Martin Chroust

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Poděkování

Na tomto místě bych rád poděkoval svému vedoucímu bakalářské práce, panu Ing. Pavlu Moravcovi, Ph.D, za to, že mi dodával užitečné rady, návrhy, a že si na mě i v těch nejméně vhodných chvílích dokázal udělat čas.

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 6. května 2011

.....

Martin Chroust

Prohlášení zástupce spolupracující právnické nebo fyzické osoby

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě dne 1. května 2011

.....

Ing. Pavel Kořenek, Ph.D.

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Martin Chroust**
Studijní program: B2647 Informační a komunikační technologie
Studijní obor: 2612R059 Mobilní technologie
Téma: Absolvování individuální odborné praxe
Individual Professional Practice in the Company

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: KVADOS, a.s.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Pavel Moravec, Ph.D.**

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Abstrakt

Bakalářská práce zahrnuje průběh odborné praxe autora ve společnosti Kvados, a.s. Hlavním pracovním zaměřením byly mobilní technologie, tedy zmapování některých současných mobilních platforem, možnost vývoje mobilní aplikací na těchto platformách v společnosti Kvados, a.s. včetně možnosti portace stávajících aplikací, a podíl na vývoji aplikace pro mapování terénních pracovníků, která byla vyvíjena na platformě Google Android. Práce byla orientována do více směrů, což koresponduje se zaměřením praxe, které bylo extenzivní a zahrnovalo velké rozpětí pracovních úkonů.

Klíčová slova

Chytrý telefon, kapesní počítač, vývoj aplikace, mobilní operační systémy, Android

Abstract

The bachelor thesis describes author's professional practice in the Kvados company. The main area of interest was mobile technologies, namely the mapping of the mobile platforms state-of-the-art, the possibility of mobile application development on selected mobile platforms and their suitability for the Kvados company, as well as the cooperation on design and implementation of field workers monitoring and management application, which was developed from scratch for the Google Android platform. The amount of work was extensive and corresponds to the professional practice focus which included a wide range of work tasks.

Keywords

Smartphone, PDA, application development, cell phone operating systems, Android

Seznam použitých symbolů a zkratek

ADB	- Android Debug Bridge
BT	- Bluetooth – bezdrátová komunikační technologie
EDA	- Enterprise Digital Assistant, kapesní počítač do terénu
EDGE	- Enhanced Data rates for GSM Evolution, mobilní datová služba
GIT	- GeoInformační Technologie
GPRS	- General Packet Radio Service, mobilní datová služba
IMSI	- International Mobile Subscriber Identity, jednozn. identifikace SIM
IMEI	- International Mobile Equipment Identity, jednozn. identif. telefonu
KML	- Keyhole Markup Language – aplikace XML v GIT
PDA	- Personal Digital Assistant, kapesní počítač
SDK	- Software Developer Kit – soubor vývojářských nástrojů
UI	- User Interface = uživatelské rozhraní (např. aplikace)
UMTS	- Universal Mobile Telecommunication system, mobilní datová služba
URL	- Uniform Resource Locator, definuje webovou adresu dokumentu
XML	- Extensible Markup Language, obecný značkovací jazyk

Obsah

1.Úvod	1
1.1 Odborné zaměření společnosti	1
1.2 Časové rozložení praxe.....	1
1.3 Seznam úkolů zadaných v průběhu praxe.....	2
2.Řešené úkoly	3
2.1 Seznámení se s aplikací myAvis.....	3
2.2 Android pro IT-Cluster, testování Samsungu Galaxy S.....	4
2.3 Systémové informace o Windows Mobile 6.5/CE	5
2.4 Technologie Adobe FLEX	5
2.5 Stahování dat ze serveru na ostravské pobočce Sazky.....	6
2.6 Hledání informací a sepisování dokumentu o možném vývoji pro Symbian	6
2.7 Testování Motoroly ES400 a chybovosti aplikace myAvis.....	7
2.8 Mobilní geoinformační systémy.....	7
2.9 myAvis pro Android – I. etapa	9
2.10 myAvis pro Android – II. etapa	15
2.11 myAvis pro Android – III. etapa	15
2.12 myAvis pro Android – IV. etapa	19
3.Potřebné znalosti získané během studia potřebné pro výkon stáže.....	23
4.Scházející znalosti či dovednosti pro absolvování praxe.....	24
5.Závěr.....	24

1. Úvod

Mobilní technologie lze dnes spatřovat ve všech odvětvích moderního života. Bez mobilních zařízení si neumíme představit běžný život, podobně to vypadá u velkých společností. Z běžného mobilního telefonu se stává pracovní nástroj, komunikátor do terénu pro sběr dat, zařízení pro monitoring a navádění zaměstnanců. Praxi ve firmě zabývající se mobilními technologiemi jsem zvolil proto, že jsem si chtěl rozšířit své mobilní obzory. Již sedm let se intenzivně věnuji testování mobilních zařízení, takže jsem od praxe očekával pohled na mobilní zařízení z druhé strany, než jen z dosavadní uživatelské.

1.1 Odborné zaměření společnosti

Tato bakalářská práce se věnuje problematice mobilních technologií a výzkumem s tím spojeným pro možné budoucí projekty společnosti Kvados, a.s., popř. pro sdružení IT-Cluster. Dále je podstatná část věnována spolupráci na vývoji aplikace myAvis pro Android, což je aplikace pro monitorování pracovníků v terénu.

Samotné pracovní úkony byly vykonávány ve společnosti Kvados, a.s. sídlící v Ostravě-Mariánských Horách. Společnost Kvados, a.s. je významným a respektovaným středoevropským producentem a dodavatelem vlastních softwarových řešení pod označením Ventus, myAvis, myCash, myWork, myTeam a myDatacenter. Na trhu působí již od roku 1992 a zaměřuje se na klienty ze segmentu obchodu a služeb. Do vývoje každý rok investuje 15% z obrátu, společnost je aktivní v 11 zemích v Evropě. Společnost je dále zakladatelem IT Clusteru - sdružení, které je vůdčí silou výzkumu a vývoje nových technologií v Moravskoslezském kraji.

Při nástupu na praxi jsem byl jako stážista zařazen k programátorskému týmu aplikace myAvis, s programátory jsem spolupracoval po celou dobu pracovní činnosti. Nejprve při zjišťování informací a poskytování rad, později při konzultacích a zadáních dalších pracovních úkonů.

1.2 Časové rozložení praxe

Semestr	Počet odpracovaných dní	Náplň práce
Zimní 2010	20	teoretická
Letní 2011	30	praktická

1.3 Seznam úkolů zadaných v průběhu praxe

V následující tabulce jsou zobrazeny zadané úkoly v průběhu praxe i s její časovou náročností:

Zadaný úkol	Časová náročnost
Seznámení se s aplikací myAvis a aktuálně nabízenými zařízeními a příslušenstvím.	2 pracovní dny
Testování Samsungu Galaxy S, instalace vývojového prostředí (SDK) pro Android, sepisování dokumentu o operačním systému Android.	5 pracovních dní
Systémové informace o systému Windows Mobile 6.5/CE.	1 pracovní den
Zjišťování informací a podrobností o sadě technologií FLEX. Sepsání dokumentu pro možné budoucí využití této technologie.	2 pracovní dny
Stahování dat ze serveru v ostravské pobočce Sazky.	1 pracovní den
Hledání informací a sepisování dokumentu o možném vývoji pro Symbian.	1 pracovní den
Testování Motoroly ES400, sepisování recenze, hledání chybových stavů na aplikaci myAvis pro Windows Mobile 6.5.3. Protokol o testování aplikace předán programátorům.	2 pracovní dny
Shromažďování informací o mobilních geoinformačních systémech, kompletní informace, funkčnost a ceny pro možné budoucí využití.	5 pracovních dní
Školení ArcGIS ve společnosti VARS Brno.	1 pracovní den
Zahájení práce na projektu myAvis pro Android – tvorba základní login obrazovky, hlavního menu, integrace a práce s GPS, reverzní geocoding, UI pro psaní zprávy, custom ListView pro zobrazení generovaného seznamu položek, integrace kalendáře, propojení s aplikací v oblasti připomenutí událostí z kalendáře. Základní logika ovládání aplikace.	7 pracovních dní
Rozšíření designu aplikace o soubory pro různá rozlišení telefonů (na testování modely Samsung Galaxy Tab, Galaxy Mini, Galaxy Ace), autoupdate aplikace přes HTTP connection, úprava grafiky a ikon.	8 pracovních dní
Integrace fotoaparátu a načtení vyfocené fotografie do aplikace. Integrace Google Map a zobrazení aktuální polohy. Vyhledání dostupných komponent pro Android, vytvoření služby autoupdate na pozadí.	9 pracovních dní
Šifrování XML, zjišťování možností navigace přes integrované Google mapy, přidání tlačítka pro okamžitou aktualizaci	6 pracovních dní

2. Řešené úkoly

Ze začátku byla moje pracovní náplň směřování k psaní dokumentů, které poukazovaly na možné budoucí využití některých technologií pro aplikaci myAvis. V další části pak následovala samotná implementace aplikace pro operační systém Android, která byla vytvořena ve čtyřech etapách, z nichž každá byla prezentována a obhájena vývojovému oddělení společnosti Kvados.

2.1 Seznámení se s aplikací myAvis

Cílem prvních pracovních dní bylo seznámení se s dosavadními přístroji a technologiemi myAvis. Bylo mi zapůjčeno aktuální i starší portfolio smartphonů, PDA a EDA. Úvodem je třeba si zmínit rozdíly mezi těmito zařízeními.

Smartphone – neboli chytrý telefon, je mobilní zařízení, které nabízí oproti běžným telefonům pokročilé funkce, určitá rozšíření a také otevřený operační systém, což znamená, že jde systém v určitých mezích upravit a je do něj možno instalovat aplikace přímo pro něj určené. Chytré telefony se dále dělí na zařízení s dotykovým displejem a bez něj. Smartphony ve firmě disponovaly operačním systémem Windows Mobile 5 – 6.1, protože stávající aplikace myAvis pro vedení pracovníků v terénu byla optimalizována a vyvíjena právě na operačním systému Windows Mobile a rezistivních dotykových displejích ovládaných stylusem (tj. dotykovým perem). Mobilní trh v oblasti smartphonů dnes čítá zhruba pět velkých operačních systémů – Android, Symbian, iOS, Windows Phone a Bada.

PDA – neboli Personal Digital Assistant je ve svém důsledku smartphone, který je zbaven telefonních funkcí. Pro připojení k internetu se využívá kabelové spojení přes ActiveSync nebo infraport. Jinak pro něj platí to samé, jako pro smartphone. Využití tohoto zařízení pro business je v dnešní době již na ústupu, protože se společnosti snaží integrovat business aplikace do zařízení, která mají pro zaměstnance širší využití.

EDA – neboli Enterprise Digital Assistant je mobilní zařízení většinou robustních rozměrů, které je určeno pro pracovníky v terénu. Zařízení jsou odolná, většinou se stylusem připevněným ke krytu přístroje provázkem nebo zacvaknutím do krytu, v zadní části se pak nachází přezka pro navlečení zařízení na dlaň. Zřídka se ještě pod displejem vyskytuje klávesnice, alfanumerická nebo QWERTY.

Při testování zařízení jsem zjistil, že u Windows Mobile zařízení je funkčnost aplikace myAvis značně roztroušená na jednotlivé modely. Každé mobilní zařízení totiž potřebuje svojí verzi aplikace pro to, aby fungovaly periferie, tj. např. fotoaparát nebo Bluetooth na vybraných COM portech. To z mého pohledu může do budoucna značně ztěžovat další implementaci na nová zařízení. V průběhu testů jsem si vyzkoušel pořizování fotografií do aplikace nebo spojení a tisk výkazů na Bluetooth tiskárně. Pochopil jsem princip fungování aplikace myAvis a prozkoumal možnosti nabídek.

2.2 Android pro IT-Cluster, testování Samsungu Galaxy S

První pracovní náplní byl operační systém Android. Byl mi zapůjčen chytrý telefon Samsung Galaxy S, abych jej otestoval a poznatky z testování zanesl spolu s informacemi o operačním systému Android do dokumentu, který bude zařazen do dokumentu IT-Clusteru „Mobilní operační systémy“. Cílem testování bylo zjistit, zda je systém Android do budoucna perspektivní pro další generaci aplikací myAvis.

Operační systém Android je systém určen pro široké spektrum zařízení, od těch nejméně vybavených až po high-end přístroje. Android staví na základním konceptu open-source. Systém si mohou výrobci stáhnout na webu a implementovat jej do vlastního přístroje zdarma. Všichni výrobci si ale systém upravují podle sebe, přidávají nové nabídky, design, grafické nadstavby a jejich povinností je tyto inovace přinést zpět komunitě Androidu. Systém Android je vlastněn společností Google, která v roce 2005 zakoupila malou společnost Android Inc. sídlící v Kalifornii. V roce 2008 bylo přestaveno první zařízení s operačním systémem Android, od té doby popularita systému prudce stoupá. Původně neznámý operační systém má dnes market-share 15% a prognózy mají rostoucí tendence. Podle agentury Gartner [5] se v roce 2014, tedy šest let po svém vzniku, dostane Android se 30% podílu na trhu do čela mobilních operačních systémů.

V dokumentu byla dále popsána architektura systému [6], nad linuxovým jádrem se nacházejí čtyři vrstvy – nativní knihovny, běhové prostředí, aplikační framework a samotné aplikace. Google pro systém Android vydává pravidelné updaty, díky již zmíněným úpravám výrobců ale jejich implementace pro konkrétní modely trvá delší dobu. Dochází tak k situacím, kdy při vydání nejnovější verze ještě ani většina telefonů nemá verzi předchozí. Jediný model však má updaty hned po vydání, je jím Google Nexus S, telefon prodávaný Googlem ale vyrobený Samsungem. Další část dokumentu byla zaměřena na integraci aplikací do systému, modelový popis uživatelského rozhraní na Samsungu Galaxy S a na potenciální náročnost provádění základních úkonů z uživatelského pohledu.

Pro vývoj Androidu se používá SDK od Googlu, které lze implementovat do vývojového prostředí Eclipse. Vývoj probíhá v Javě a návrh GUI v XML. XML totiž definuje designovou strukturu, zatímco Java se stará o samotný kód aplikace. Každá zobrazená obrazovka se pak rovná jedné aktivitě, ve které je navázán kód programu i XML soubor. Podle prvních poznatků Android vyšel z práce poměrně dobře. Jeho nespornou výhodou je adaptace aplikací na různá rozlišení displejů, stačí jen upravit velikosti vkládaných obrázků a grafiky, popř. upravit layouty (tj. rozložení ovládacích prvků na displeji) jednotlivých aktivit. Prohlídka uživatelského rozhraní Androidu ukázala, že by aplikace myAvis pro Android měla do budoucna šanci.

Celý dokument je umístěn v Příloze A.

2.3 Systémové informace o Windows Mobile 6.5/CE

Dalším pracovním úkonem bylo vyplnit dokument o operačním systému Windows Mobile 6.5 z hlediska vývojového prostředí, bezpečnosti, databázových predispozic, uživatelského rozhraní, komunikace, podpory periférií, zálohování, deploymentu a instalace aplikací, vzdálené správy, power managementu, dostupného softwaru třetích stran, vlastností systému a softwaru pro správu.

Vycházel jsem z vlastních zkušeností z dřívějšího testování smartphonů s operačním systémem Windows Mobile a z rozhovorů s programátory z oddělení myAvis. Výsledný dokument se nachází v Příloze B, po zhotovení byl odeslán na VŠB-TU Ostrava jako referenční zadání pro zhotovení obdobných statistik i pro jiné operační systémy s cílem je mezi sebou ve výše uvedených bodech navzájem porovnat.

2.4 Technologie Adobe FLEX

Další teoretická náplň práce spočívala ve zjištění podrobností o platformě Flex. Byla zde možnost budoucího využití této technologie pro některé interní testovací projekty společnosti Kvados, a.s.

Adobe Flex [8] je sada technologií od Adobe Systems pro vývoj a nasazení platformě nezávislých interaktivních internetových aplikací založených na Adobe Flash. Podporuje-li tedy daný prohlížeč nebo operační systém technologii Flash, je možné v něm rovněž spustit aplikace založené na Adobe Flex. Jaké z toho plynou výhody? Flex dokáže běžet nezávisle na vybraném prohlížeči, tedy vždy stejně. Potenciál tohoto řešení pro možné budoucí směřování mobilních aplikací je ve velkém rozšíření Flash pluginu napříč různými systémy, má jej až 95 % počítačů. Stejně jako u Androidu, tak i u FLEXu došlo k oddělení struktur – MXML se používá pro tvorbu uživatelského rozhraní, zatímco ActionScript 3.0 slouží k psaní procedurálního kódu. Vývoji může napomoci velká komunita vývojářů i komponenty třetích stran, popř. existence frameworků. Framework je z programátorského hlediska užitečný z rozdělení kódu na dílčí části. Nejznámějším příkladem je Model-View-Controller, zkráceně MVC. Základ spočívá v rozdělení programu na část dat (Model), programovou část starající se o vzhled (View) a komponentu Controller, která se stará o interakci mezi Modelem a View.

Flex má však i řadu restrikcí, ke kterým patří nemožnost čtení a zápisu na straně koncového uživatele, Flash běží v sandboxu a nejde se přes něj dostat k souborovému systému. K MySQL databázi se nejde rovněž připojit, takže pro interakci s daty zbývá jen HTTP připojení pro vyžádání dat ze serveru. Pro tyto restrikce není FLEX v aktuální fázi rozpracování příliš vhodný. Nabízí jisté validní body, interakce s daty a databází je však pro aplikaci myAvis klíčová. Do té doby, než budou změněny restrikce nebo zlepšena integrace dat, je Flex pro myAvis nepoužitelný.

Dokument popisující technologii Adobe FLEX se nachází v Příloze C.

2.5 Stahování dat ze serveru na ostravské pobočce Sazky

Součástí mé praxe ve firmě bylo i stahování dat ze serveru na ostravské pobočce Sazky. Mým úkolem bylo dohlížet na bezproblémový přenos dat ze serveru do mobilní a Windows CE verze aplikace myAvis a v případě chyby stahování tento proces obnovit. Data byla úspěšně stažena a odvezena zpět do sídla společnosti Kvados, a.s.

2.6 Hledání informací a sepisování dokumentu o možném vývoji pro Symbian

Dále byla zjišťováno, co je potřeba pro vývoj pro operační systém Symbian. Systém jako takový je velmi roztroušený, dělí se na nedotykové verze a dotykové – v době psaní dokumentu to byly – Symbian^3, Symbian S60 5th edition a Symbian S60 3rd edition. Pro vývoj na Symbian je tedy potřeba nejdříve si rozmyslet, jakou verzi zvolit. Protože nejsou implementace mezi dotykovou a nedotykovou verzí navzájem kompatibilní, musela by se vytvořit dvojice aplikací, každá s úplně jiným uživatelským rozhraním a ovládáním.

Pro nejnovější verzi systému - Symbian^3 jsou používány nástroje WRT (Web RunTime) pro tvorbu widgetů, tj. miniaplikací, které se zobrazují na obrazovce. Nejlépe je začít s programem Nokia Qt SDK [9], v němž jsou obsaženy všechny základní prvky, je navíc multiplatformní, takže jde spustit na Windows, Linuxu nebo Mac OS X. Pro pokročilejší programování je k dispozici program Carbide.c++, poradí si s Qt knihovnami a navíc je již od roku 2008 distribuován zdarma. Na platformu Symbian jde dále vyvíjet v Pythonu nebo Flashi. Pro Flashové aplikace je potřeba program Flash Professional CS5, python jde vyvíjet ve vlastní konzoli v počítači nebo po doinstalování překladače i v konzoli přímo na mobilních zařízeních.

Jakmile je aplikace u Symbianu vyvinuta, je třeba ji nějakým způsobem dostat na trh. Aplikace ve formátu SIS musí být před svoji distribucí podepsány, aby bylo možné využívat přístup k určitým potřebným funkcím hardwaru a softwaru. V případě nekomerčních aplikací jde aplikace podepsat zadarmo na přesný IMEI telefonu. Tento krok má za následek zobrazení zprávy informující o vývojové verzi aplikace po jejím spuštění. Pro komerční účely je potřeba TC Publisher ID (200 USD za rok) a navíc se za každé schválení účtuje poplatek 560 EUR. Pokud je aplikace vrácena pro nedodělky k přepracování, je znovu navíc účtováno 280 EUR. To může potenciální vývojáře odrážet, dalším negativním faktorem je složitost celé operace podepsání, která se může protáhnout až na několik týdnů. Na trhu však existuje řada subjektů, které nabízejí podepsání téměř bez námahy.

Dokument o vývoji pro Symbian se nachází v Příloze D.

2.7 Testování Motoroly ES400 a chybovosti aplikace myAvis

V listopadu dorazila do Kvadosu Motorola ES400, která se má stát hlavním smartphonem pro aplikaci myAvis na operačním systému Windows Mobile 6.5. Počet smartphonů s operačním systémem Windows Mobile 6 rapidně klesá (už se téměř nevyrábí, nejnovější smartphony od Microsoftu se prodávají s operačním systémem Windows Phone 7), a když už se některý objeví, nabízí pouze kapacitní displej, který je uzpůsoben na ovládání prsty. Aplikace myAvis však potřebuje stylus a s tím spojený rezistivní displej, Motorola ES400 je tedy jeden z nejaktuálnějších modelů, který vyhovuje displejem, odolnými kryty pro pohyb v terénu, funkční nabídkou i přijatelnými rozměry. Součástí pracovní náplně tedy bylo smartphone vyzkoušet, zjistit některé užitečné možnosti, které mohou potenciální uživatelé ocenit, a sepsat krátkou recenzi, která se nachází v Příloze E.

V návaznosti na recenzi mi byl udělen úkol vypomoci testerům a zjišťovat chybové stavy v nejnovějším buildu aplikace pro Motorolu ES400. Nalezl jsem devět systémových chyb týkajících se grafiky, problémů s databázovým předáním hodnot mezi tabulkami a pádem aplikace s neočekávanou chybou. Vše jsem zanesl do Protokolu o testování aplikace, který byl odeslán programátorům.

2.8 Mobilní geoinformační systémy

Další částí výzkumu možného budoucího vylepšení stávajících aplikací jsou mobilní geoinformační systémy. Mým úkolem bylo vypracovat zprávu to obecných informacích o mobilních GIS, popsat jednotlivé části, obeslat dodavatele GIS systémů v České republice s případnými dotazy a vše spolu s nalezenými GIS přístroji zanést do dokumentu.

Geoinformační systémy [11] jsou všeobecně nějaké informační systémy, které editují, integrují, analyzují, sdílí a zobrazují geografické informace. GIS aplikace jsou nástroje, které umožňují uživatelům vytvářet dotazy nad daty, analyzovat prostorové informace, editovat mapy a data a prezentovat výsledek těchto operací. Pro potřeby mobillingu mě ale nejvíce zajímaly mobilní GIS, což je spojení GIS s mobilními zařízení. Hlavní úlohy mobilních GIS jsou:

1. **prohlížení map a navigace** – přidáním podpory GPS navigace vznikají aplikace, kdy mobilní uživatelé mohou reportovat své souřadnice vedení = mapování pracovníků v terénu
2. **inspekce v terénu** – např. skupiny pracovníků v terénu provádějí inspekční činnost a do aplikace reportují provozní stav, často i fotografii objektu a zadají jej s přesnou souřadnicí do mapy jako zkontrolovaný objekt. Nedochozí k přidání nových objektů ale jen upgradů těch stávajících
3. **sběr dat v terénu** – získávání nových dat podle typu objektů a jejich pozic, základní typy dat – bod a polygon

Mobilní GIS je používán na běžných smartphonech, dále pak na odolných smartphonech, odolných smartphonech s klávesnicí, tabletech a vysoce přesných GPS zařízeních. Většinou se jedná o aplikaci, která se stará o sběr dat. V současné době na trhu existuje řada firem distribuujících GIS software, např. Autodesk, ESRI, Integrator, Mapinfo, GE Smallworld a další. Mezi českými firmami věnující se GIS najdeme např. Berit, DIGIS, Gepro nebo Xanadu. Pro myAvis byl vybrán asi nejznámější a světově nejpoužívanější GIS řešení od ESRI. Distributory v České Republice jsou ARCDATA Praha nebo VARS Brno. ESRI nabízí aplikace určené pro mobilní GIS - ArcGIS Mobile a ArcPad.

ArcGIS Mobile

ArcGIS Mobile [12] je dostupný v hotové verzi nebo jako SDK, což může sloužit pro inkorporaci geo-části do businessově laděných aplikací. ArcGIS Mobile jde účelově upravit tak, aby nabízela přesně to, co se od koncového pracovníka v terénu očekává. Tvorba projektů probíhá v Mobile Project Center, kde lze manuálně přidávat dotazy nebo upravovat funkčnost. Jde tedy nastavit co uživatelé uvidí, co budou moci editovat a změnit popisky všech činností. Protože je aplikace jinak celá v angličtině, jde ji touto cestou přeložit do češtiny. Mezi možné druhy úkolů aplikace se řadí prohlížení dat, sběr dat v terénu, tvorba dotazů (co a kde hledat, dotazy mohou být i složené), pracovní seznam (výpis věcí, na které nesmí pracovník zapomenout), prohlížení úprav a capabilities, tj. přihlášení do aplikace a sledování dalších pracovníků. Pracovníci mají přístup pouze k volbám, ke kterým je jim umožněn přístup, což je bezpečnější pro méně zkušené uživatele. Informace o ArcGIS byly získány z jednodenního školení ArcGIS Mobile ve společnosti VARS Brno, a.s. - viz příloha F.

ArcPad

Zatímco ArcGIS Mobile je plně přizpůsobitelná aplikace, ArcPad je úplným opakem. Jedná se o aplikaci pro desktopové i mobilní systémy Windows. Výhodou je to, že má aplikace stejné uživatelské rozhraní v mobilu i na počítači. Uživatel si může projekt předem připravit, nastavit a po nahrání do zařízení hned může začít mapovat. ArcPad je oproti ArcGIS Mobile aplikací spíše pro pokročilejší uživatele, najdeme zde mnoho voleb a nastavení, včetně možnosti zanesení nových dat do mapy. Je možno zaznamenávat body, linie, polygony a další prvky. Při zaznamenávání nových bodů jde vyplňovat potřebné atributy, samotné body se ukládají do předem vytvořených vrstev.

V závěru dokumentu o GIS byla uvedena i některá ukázková zařízení pro GIS aplikace, leaderem na trhu je společnost Trimble, specializovaná zařízení s odolnými kryty si lze na objednávku pořídit za částky v řádu desítek tisíc Kč.

Ve společnosti Kvados, a.s. je v současné době zvažována možnost propojení aplikace myAvis s GIS řešeními společnosti ESRI. Celý dokument o Geoinformačních systémech se nachází v Příloze G.

2.9 myAvis pro Android – I. etapa

Stěžejní náplní mé praxe u společnosti Kvados, a.s. bylo programování aplikace myAvis pro mobilní operační systém Android. Ve společnosti běžel interní projekt ohledně aplikace pro Android spolu s analýzou, kde byla nastíněna funkčnost prvotní verze aplikace, avšak nikde nebylo definováno uživatelské rozhraní aplikace. To bylo mým prvním pracovním úkonem, tedy dát aplikaci jistý vzhled a organizovanou strukturu s jasně vymezeným ovládáním.

Na úvod v jednoduchosti ozřejmím některé pojmy, které budu v dalších sekcích bakalářské práce používat:

Activity – jedná se o aktuální uživatelské rozhraní, tj. vše to, co je v dané chvíli zobrazeno na displeji telefonu. Aplikace pro Android může definovat řadu aktivit, z nichž každá je zodpovědná za uložení svého stavu při svém ukončení tak, aby se k ní v rámci životního cyklu aplikace dalo opět vrátit.

Intent – mechanismus pro určité specifické akce, např. „odešlí e-mail, vyfoť fotografii“. Když chceme provázat stisknutí tlačítka na nějakou akci, využíváme některý z přednastavených intentů. Pokud chceme udělat interaktivitu mezi vlastními naprogramovanými aktivitami, vytváří se nový intent, kterému jako parametr předáváme další aktivitu.

Service – jedná se o úlohu, která běží na pozadí bez toho, aby po celou dobu interagovala (tj. vizuálně komunikovala) s uživatelem. Android má řadu systémových služeb běžících na pozadí, jdou ale vytvořit i vlastní a delegovat jejich trvání, např. spuštění po startu aplikace, zrušení při jejím ukončení, apod.

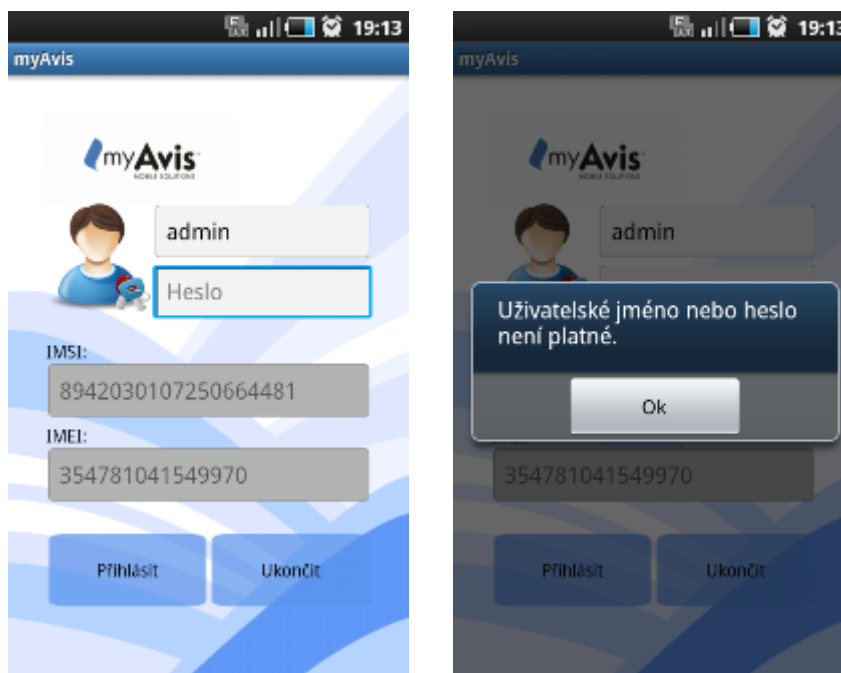
Layout – každá aktivita má svůj layout, tedy základní definici rozvržení designových prvků. Napsaný je v XML a je požadováno, aby obsahoval alespoň jeden layout. Druhů layoutů je několik a dají se vnořovat. Do layoutů se pak umísťují určité prvky uživatelského rozhraní (tlačítko, pole pro obrázek, textové pole, apod.)

Manifest – dokument přítomný v každé aplikaci Androidu, definuje název aplikace, verzi, použité knihovny, použité aktivity (bez jejich implicitní deklarace v aplikaci nefungují), intent filtry, použitou verzi SDK, uživatelská oprávnění (bez oprávnění nelze spustit určité funkce, popř. vykonat určité činnosti) a použité hardwarové prvky (bez oprávnění je opět nelze použít). Použitá verze SDK znamená minimální úroveň API, pro kterou je možno aplikace ještě použít (nová úroveň = nová verze systému Android). Nové verze mohou přinášet nové implementace, ale jsou kompatibilní i s verzemi staršími.

Vývoj jsem prováděl ve freewarové aplikaci Eclipse s Android SDK pluginem. Testování aplikace probíhalo v integrovaném emulátoru systému Android 2.1 a také v mobilním telefonu Samsung Galaxy S s Androidem 2.2. Některé ikony hlavního menu byly staženy z internetu s licencí „*free for commercial use*“, další grafika byla nakreslena v grafickém programu GIMP.

Přihlašovací obrazovka

U úvodní obrazovky (*main.java* a *main.xml*) je jako hlavní použit *RelativeLayout* s vnořenými *LinearLayout*. Relativní byl jako hlavní použit z důvodu nerovnoměrného rozvržení prvků – obrázku vzhledem k zadávacím polím. První dvě textová pole slouží pro zadání uživatelského jména a hesla. V první poli se zadaná hodnota ukládá, aby při příštím spuštění mohla být znovu ve stejném znění již předvyplněna. Na starosti to má metoda `getSharedPreferences`. Heslo je vyplňováno hvězdičkami (resp. tečkami) díky nastavení vlastnosti `android:password="true"` u prvku `<EditText>` v XML souboru. Aplikace jako taková nemá být distribuována přes centrální obchod s aplikacemi Android, tj. Market, ale bude dodávána nezávisle firmou Kvados, a.s. Z tohoto důvodu je třeba pro prvotní registraci uživatele na vzdálený server zaslat hodnotu čísel IMSI a IMEI, která zaručují jednoznačnou identifikaci. IMSI je jedinečné číslo SIM karty, IMEI jedinečné číslo telefonu. V budoucích verzích aplikace budou tyto hodnoty pravděpodobně ukryty. IMEI je získáván přes instanci služby *TelephonyManager* `telephonyManager.getDeviceId()` a jednoznačné identifikační číslo SIM karty IMSI jako `telephonyManager.getSimSerialNumber()`. Ošetření hesla provádím příkazem *if-else*, zatím kontrolovat obsah polí, zdali je shodný s pevně nastaveným jménem a heslem. V budoucnu bude programátory rozšířen o funkcionalitu napojení na databázi firemního serveru, kde se bude uživatel přihlašovat. V současné fázi je při špatném zadání vypsána tabulka s chybovou hláškou.



Obrázek 1: a) Přihlašovací obrazovka b) neplatné zadání uživatelského jména

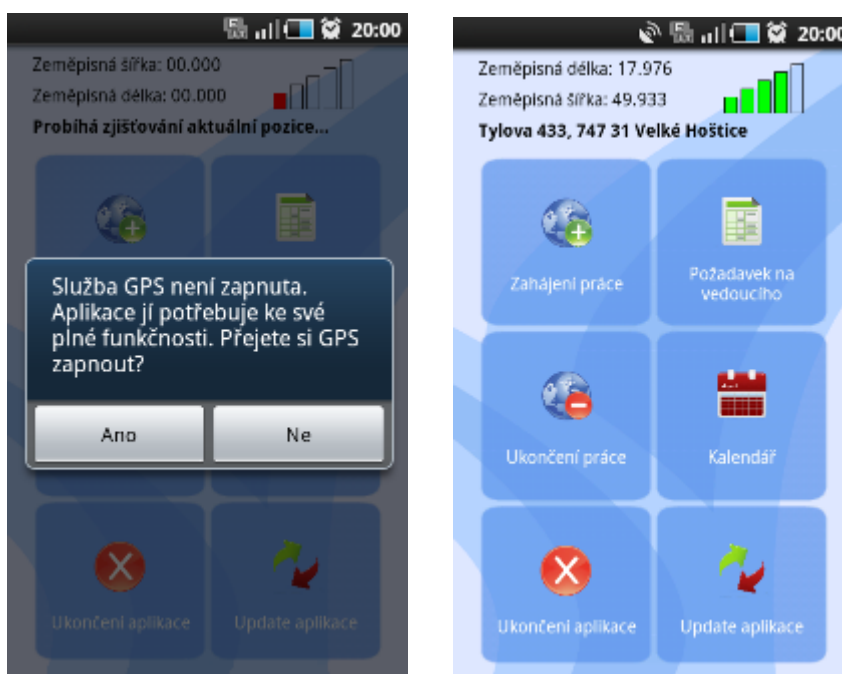
Ve spodní části jsou umístěna dvě tlačítka, tlačítko Přihlásit po úspěšné verifikaci údajů otevře hlavní menu, tlačítko Ukončit aplikaci ukončí. Tlačítka nejsou defaultní, ale mají vzhled podle obrázku, který je na jejich pozadí. Systém Android má i některé povinné hardwarové klávesy,

takže bylo třeba upravit i jejich chování. Zatímco prostřední klávesa pro zobrazení plochy je programátorsky nedostupná, na zpětnou šipku jsem byl po stisknutí naprogramován dotaz, zdali chce uživatel skutečně aplikaci ukončit. Při náhodném stisknutí, zvláště u senzorových kláves, tak nedojde k okamžitému ukončení aplikace.

Hlavní menu

Při njetí do hlavní nabídky tímto kódem zjišťuji, zdali je aktivní GPS připojení:

```
manager = (LocationManager) getSystemService( Context.LOCATION_SERVICE );  
if ( !manager.isProviderEnabled( LocationManager.GPS_PROVIDER ) ) {  
    alertMessageNoGps();  
}
```



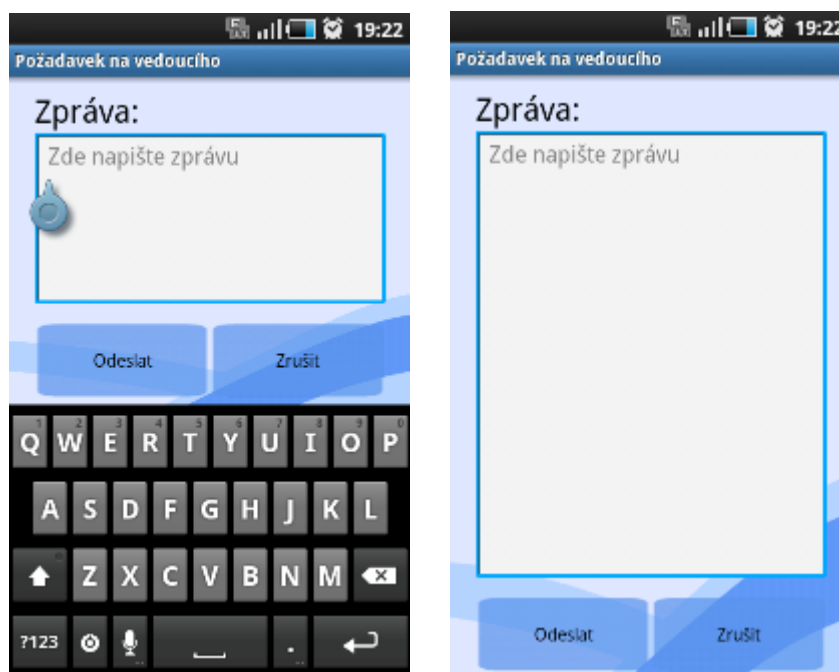
Obrázek 2: a) Vynucení zapnutí GPS b) Hlavní menu aplikace

V případě, že GPS není aktivní, zobrazím hlášku `alertMessageNoGps()`, která uživatele upozorní na to, že funkce GPS je pro danou aplikaci klíčová, a že by si ji měl po kliknutí na tlačítko Ano zapnout. Po stisknutí se objeví obrazovka *Informace o poloze a zabezpečení*, kde lze zapnout použití satelitů GPS i bezdrátových sítí. V horní části menu se totiž nachází místo pro zobrazení souřadnic, tj. zeměpisné šířky a zeměpisné délky. Obě hodnoty mají pro přesnost pokrytí zemského pokrytí desetinnou částí o délce čtrnácti cifer, pro úsporu místa však byla celá hodnota omezena na maximálně šest znaků (počítají se číry, desetinná tečka a záporná hodnota). Vpravo od souřadnic se nachází měřič přesnosti GPS signálu podle satelitů, od kterých telefon přijímá satelitní data.

Přesnost je určována podle parametru `location.getAccuracy()`, tj. přesnost GPS signálu, která je udávána od 0.0 (signál nenalezen) a výše. Pod zeměpisnými souřadnicemi se nachází řádek, kde se zobrazuje aktuální ulice s městem podle aktuální souřadnice. Využíván je k tomu tzv. „*reverzní geocoding*“ ve vláknech, aby nebylo při zjišťování zaseknuto uživatelské rozhraní. Dokud není lokace nalezena aplikace vypisuje hlášku „Probíhá zjišťování aktuální pozice...“, později zobrazí aktuální adresu. Samozřejmě je potřeba brát v potaz indikátor přesnosti GPS.

Pro hlavní menu bylo zvoleno rozvržení 3 x 2 ikony na řádku, ikony jsou ve skutečnosti tlačítka opět s volitelným pozadím, přes které jsou vykresleny PNG ikony, pod nimiž je vypsán popisik tlačítek. Hlavní menu je plánováno na pozdější rozšíření, buď se přidá tlačítko, které povede na další část menu nebo bude nabídka scrollovatelná, popř. může být na displeji místo šesti čtvercových ikon vyobrazeno až devět obdélníkových. Činnost aplikace po stisknutí tlačítek Zahájení práce a Ukončení práce bude implementována programátory ve firmě Kvados, protože se bude jednat o napojení na stávající serverové rozhraní. Funkčnost byla implementována u tlačítek Požadavek na vedoucího, Kalendář a Ukončení aplikace. Pro zobrazení tlačítek v matici byl použit Table Layout, pro zobrazení informací o GPS byl použit Relative Layout.

Požadavek na vedoucího



Obrázek 3: a) Požadavek na vedoucího s aktivovanou klávesnicí b) při zavřené klávesnici

Z pohledu rozvržení se jedná o jednoduchou aktivitu s položkou `<EditText>` přes téměř celý displej, v dolní části jsou pak vyobrazena custom tlačítka Odeslat a Zrušit. Nejnáročnější implementací je zde posun tlačítek při vysunutí klávesnice, která se objeví hned po najetí aktivitu. Tlačítka jsou zobrazena hned nad klávesnicí, čímž sice zabírají prostor pro text, ale ten uživatel

může na zbývajících řádcích stejně zadat a po dokončení textu okamžitě odeslat. Může za to nastavení parametru `android:windowSoftInputMode="stateVisible|adjustResize"` v Manifestu aplikace a také vhodně zvolený layout, tj. kombinace Relative a Linear Layoutu. Po odeslání nebo zrušení se uživatel vrátí do hlavního menu. Pokud zprávu odeslal, objeví se na displeji krátkodobá zpráva (tj. Toast zpráva) o úspěšném odeslání, která za několik sekund zmizí. Závěrečná implementace, tedy napojení odeslání zprávy na rozhraní a databázi serveru, bude řešeno opět týmem programátorů myAvis.

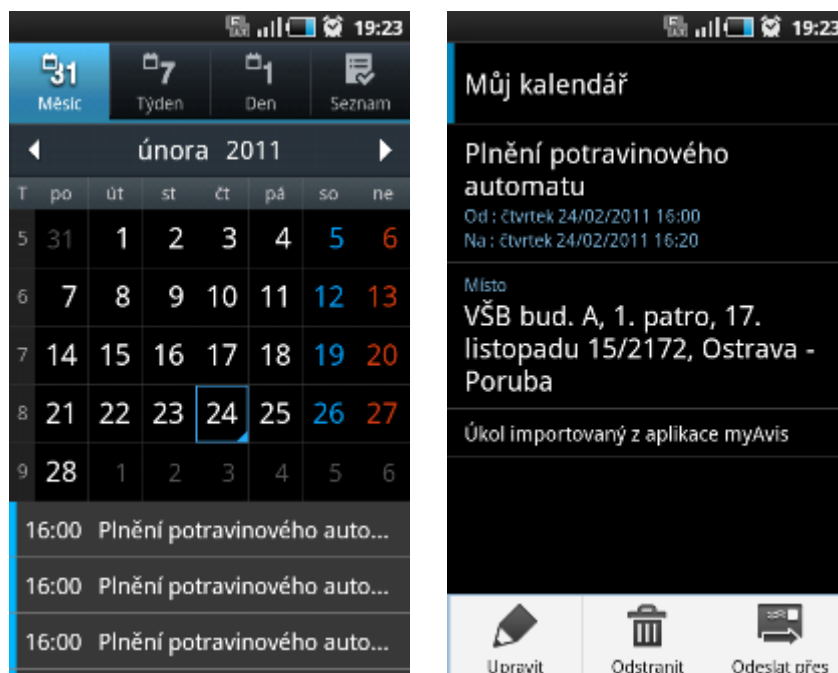
Kalendář



Obrázek 4: a) Požadavek na vedoucího s aktivovanou klávesnicí b) při zavřené klávesnici

Aktivita kalendář obsahuje plán pro pracovníka v terénu na několik dní dopředu. Z programátorského hlediska se jedná o custom ListView, tedy rolovací seznam, kde je struktura záznamů nastavena v samostatném XML dokumentu. V programovém kódu je pak nastaveno provázání dat z třídy calendar.java s listem. Pro vytváření nových objektů je manuálně vytvářena instance třídy tasks - `myTasks = new ArrayList<tasks>()` a jsou vyplněny potřebné údaje, tedy název úkolu, datum, čas začátku a konce a místo. V testu zátěže bylo vytvořeno 200 záznamů, aby se zjistilo, zdali stíhá telefon list překreslovat. Pohyb je trochu trhaný, ale to má za následek vlastnost Androidu – recyklace. Znamená to, že při listování seznamem dolů se horní položka dostane jakoby za okraj, avšak znovu se použije (tj. recykluje se) o objevivší spodní položky, jen s novými daty. Po napojení na databázi serveru se budou položky stahovat ze serveru, zobrazení probíhá ve vláknech se zobrazeným loaderem aby při velkých počtech položek a informací nedošlo

k zablokování uživatelského rozhraní. Položky seznamu jsou interaktivní, pokud na ně uživatel klikne, zobrazí se okno AlertDialog, ve kterém jsou vypsány podrobnosti o daném úkolu s možností uložení do interního kalendáře po stisknutí tlačítka Uložit. Přesné datum a čas upozornění se zadává přes DatePickerDialog a TimePickerDialog po kliknutí na tlačítka, jejichž text odpovídá aktuální nastavené hodnotě data, resp. času.



Obrázek 5: a) Kalendář s událostmi importovanými z aplikace myAvis b) detail importované události

Vložené záznamy z myAvisu v kalendáři mají nastavené všechny atributy přejaté z aplikace myAvis, navíc je přidán řádek s informací o tom, že úkol byl importován z aplikace myAvis. Záznamy lze ručně upravovat.

Ukončení aplikace

Po kliknutí na ukončení aplikace a potvrzení volby se ukončí nejen aktuální Aktivita ale i předešlá (tj. login screen), aby došlo k úplnému ukončení aplikace. Ošetřil jsem také stisknutí zpětné šipky, tady má uživatel možnost volby mezi ukončením aplikace a odhlášením, kdy se dostane zpět na přihlašovací obrazovku.

2.10 myAvis pro Android – II. etapa

V další části práce na aplikaci pro Android mi bylo zadáno přidání možnosti updatu aplikace přes HTTP server. Navíc byla aplikace předělána na různá rozlišení displejů, resp. pro různou hustotu pixelů na displeji.

Update aplikace je vytvořen v hlavním menu, spustí se po kliknutí na ikonu Update. Přes `HttpURLConnection` je nejprve zjištěna aktuální verze a velikost verze na serveru a je nabídnuto okno, ve kterém jsou tyto informace zobrazeny. Po stisknutí tlačítka Aktualizovat se přes vytvořenou Async Task začne stahovat soubor umístěný na pevné URL adrese přes HTTP spojení. Async Task se používá místo vláken, implementace je jednodušší, předpřipravená, ale jde v aplikaci použít pouze jedna. Při stahování je na displeji vidět progress bar informující v reálném čase o procentuálním poměru stažených dat proti datům zbývajícím. Aplikace se stahuje do složky `Environment.getExternalStorageDirectory()+"/download"`, která je běžně používána pro stahování z internetového prohlížeče, protože jsem nezískal přístup k zápisu do složky `myAvis/data`, což je adresář pro uživatele běžným způsobem nepřístupný. Cílem bylo, aby aplikace nebyla uživateli po stažení do telefonu přístupná. Po stažení souboru se však hned evokuje režim instalace, kde data aplikace budou zachována, takže je jedinou možností stažený soubor hned po najetí aplikace smazat.

Na testování jsem obdržel několik zařízení Samsung s modelovými hustotami pixelů na displejích. Galaxy Mini suploval nízkou hustotu (ldpi), Galaxy Ace střední (mdpi) a Galaxy S spolu s Galaxy Tab velkou hustotu (hdpi). Pokud je Android aplikace napsaná standardně pro rozlišení hdpi, jde samozřejmě spustit i na jiných displejích, avšak ne vždy jsou výsledky plně kompatibilní. V případě aplikace myAvis se jednalo o odřezání částí tlačítek u login screenu, popř. se na displej nevešly ikony hlavního menu. Proto bylo třeba vytvořit podobnou sadu layoutů pro různé displeje. Dosavadní layouty jsem vytvářel ve složce layout-hdpi, takže jsem je nakopíroval do složek layout-ldpi a layout-mdpi s tím, že bylo potřeba provést celou řadu úprav, aby se jednotlivé verze na telefonech zobrazovaly bezchybně. To stejné bylo potřeba udělat i se složkou drawable-hdpi (tedy zkopírovat ji do drawable-ldpi a drawable-mdpi a upravit), která obsahuje objekty obrázků a drží jejich referenci pro navázání obrázků do aktivit. Výsledkem byla funkční aplikace na všech testovaných zařízeních s minimální grafickými odlišnostmi.

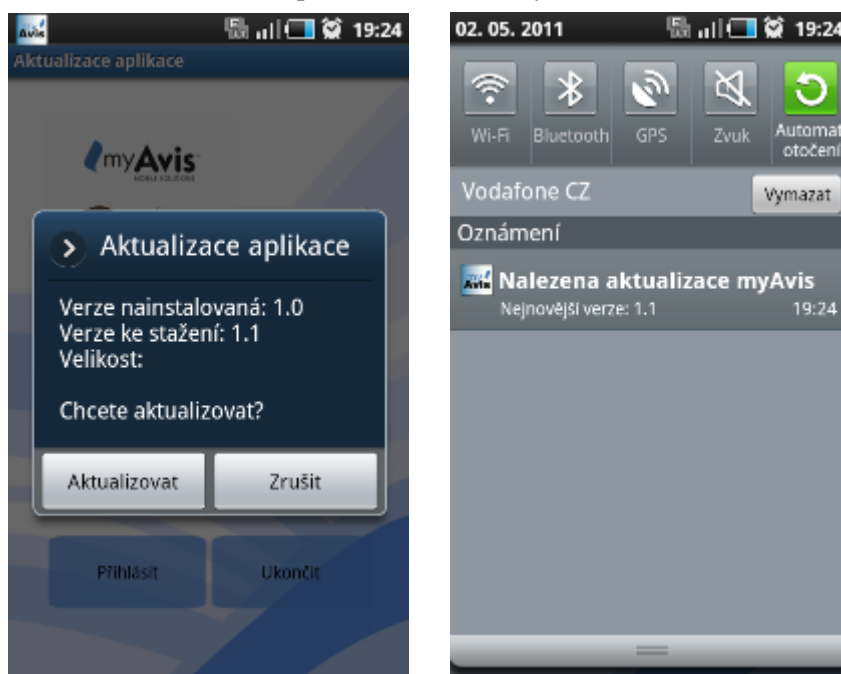
Fotogalerie vzhledu aplikace myAvis na zařízeních s různou hustotou pixelů na displeji se nachází v Příloze H.

2.11 myAvis pro Android – III. etapa

Ve třetí části vývoje byla cílem úprava aktualizace, která má běžet jako služba na pozadí. S aplikací má být distribuován konfigurační soubor, přes který půjdou načítat interní data, verze systému pro updaty a také třeba to, zdali je update povinný nebo jen doporučený. Dále bylo zapotřebí integrovat Google mapy, interakci s fotoaparátem a prozkoumat možnosti dostupnosti

placených a freewarových komponent pro operační systém Android.

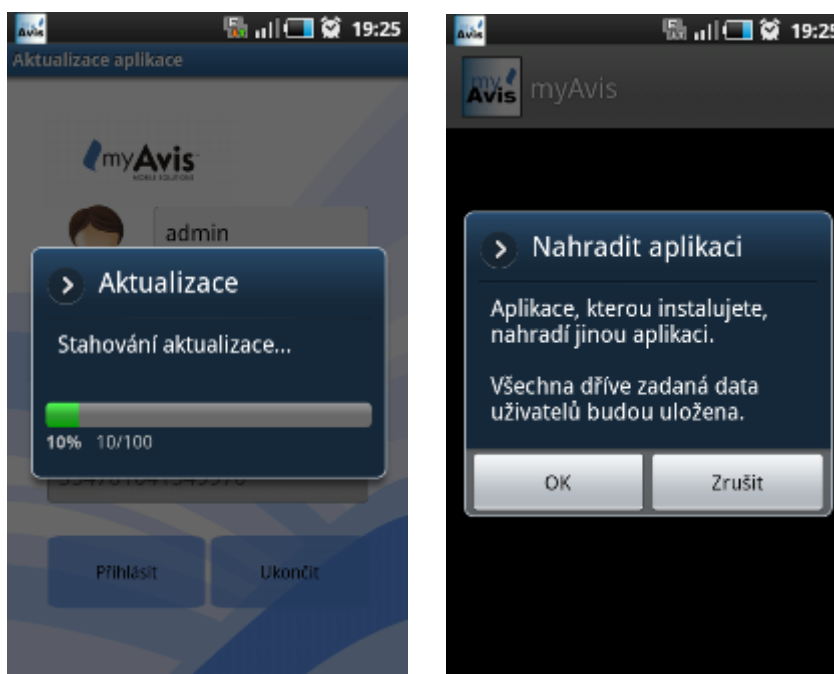
V prvé řadě bylo potřeba předělat hlavní nabídku tak, aby byla schopna pojmout nově vytvořené ikony. Z možností scrollování, rozšíření stávajícího počtu ikon na obrazovce nebo další obrazovky menu jsem zvolil střední cestu, protože šestice ikon byla na displeji zbytečně velká. Poté bylo třeba předělat dosavadní `updateAlert` na vlastní aktivitu. Protože však byla aktivita tvořena přes `AlertDialog.Builder`, musel jsem při zachování stávajícího kódu vyřešit to, aby se na obrazovce vizuálně nezobrazovala celá obrazovka nové aktivity. To jsem vyřešil nastavením průhledného tématu `android:theme="@android:style/Theme.Translucent"` v Android Manifestu. Další problém se objevil v případě, kdy uživatel soubor sice stáhnul, ale samotnou instalaci zrušil. Poté se totiž vrátil do aktivity, která byla zobrazena průhledně a vizuálně nereagovala na dotyky. Bylo třeba přepsat metodu `onActivityResult` v nově vytvořené aktivitě `update.java`, ve které se volá metoda pro ukončení aktivity.



Obrázek 6: a) Dotaz na aktualizaci aplikace b) služba informující v notificačním řádku o nové dostupné verzi

Další dílčí částí automatické kontroly nových verzích na webu bylo přidání XML souboru, který v sobě ponese konfiguraci aplikace, tj. verzi aplikace a URL pro update a kontrolu nových verzí. Soubor `config.xml` jsem uložil do složky Assets v rámci SDK, který se k aplikaci po sestavení přibalí. Aktivita update má nyní nově dva XML parsery (tj. nástroje pro rozložení XML a získání patřičných dat), jeden pro čtení z lokálního configu z Assets, druhý pak pro parsování údajů na webu. Nejprve se přečte lokální `config.xml`, ze kterého se zjistí verze aktuálně nainstalované aplikace, URL pro stažení nové verze a URL obdobného XML souboru na webu. Jestliže je parametr version na webu číselně vyšší (např. 1.1 vs. 1.0), je nutno updatovat. Samotný update je řešený službou, která je spuštěna ihned po spuštění aplikace. Služba si zkontroluje verze a pokud je

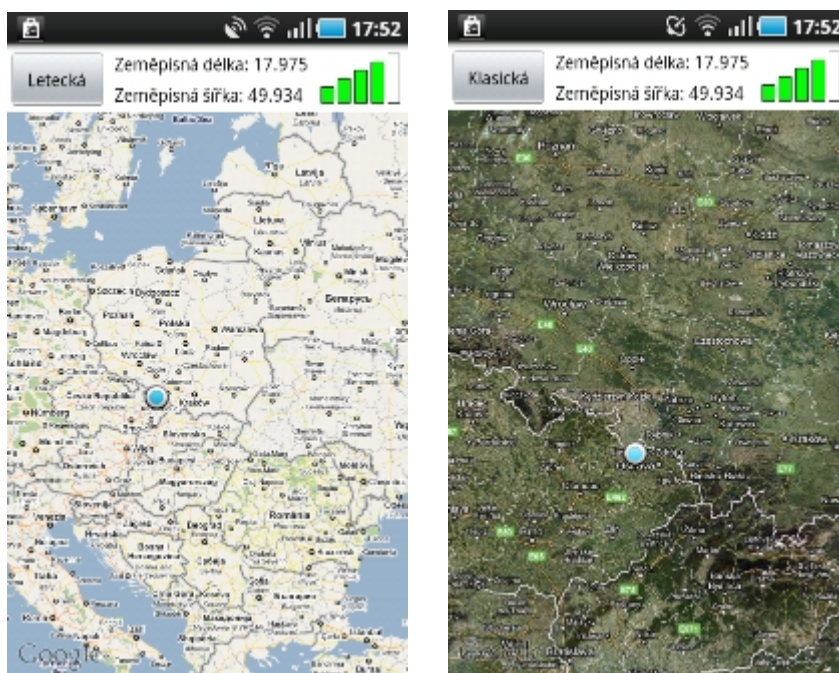
na webu vyšší, nastartuje aktivitu update a v notificačním řádku Androidu zobrazí informaci o tom, že je na webu dostupná nová verze aplikace i s jejím číselným označením. Pokud uživatel odmítne update, je v notificačním řádku stále zobrazena ikona updatu. Pokud na ní uživatel klikne (nebo klikne na výzvu v aplikaci), je aktivováno stažení souboru přes AsyncTask a jeho následná instalace. Pokud po spuštění služby není na webu aktuální verze, uživatel nic nepozná. Služba běží na pozadí a v zadaném intervalu daným časovačem (nastaveno bylo 30 minut, tj. 1 800 000 ms) pravidelně kontroluje přítomnost nové verze. V XML konfiguračním souboru je i tag *Obligatory* značící povinnost updatu. Do budoucna může být aplikace upravena tak, aby v případě, že je update povinný, nepovolila uživateli přihlášení, dokud nepotvrdí instalaci a nenainstaluje nejnovější verzi. Tím je řešena rychlá oprava chyb v aplikaci – nová verze se dostane na koncová zařízení vynuceným updatem takřka hned. Služba je ukončována při opuštění aplikace dostupnými tlačítky, při ukončení aplikace přes Správce úloh nebo při přeinstalaci celé aplikace.



Obrázek 7: a) Stahování aktualizace b) Instalace nově stažené verze

Další potřebnou součástí aplikace je integrace Google Map. Každý smartpohne s operačním systémem Android má předinstalovanou mapovou aplikaci Google Maps, která stahuje aktuální mapové podklady z internetu. Pokud chce programátor využít mapy ve své aplikaci, musí nejdříve zjistit MD5 otisk klíče, kterým podepisuje svou aplikaci. V rámci vývoje je aplikace automaticky podepisována při buildu a to univerzálním vývojářským klíčem. Pro využití map je třeba nejdříve podepsat svou aplikaci vlastním vývojářským klíčem a jeho otisk pak poslat na servery Googlu, kde je obratem odeslán zpět řetězec znaků, který je potřeba vložit do MapView a mapy budou fungovat. Od té doby ale musí být aplikace podepisována výhradně vaším novým klíčem (tj. nově vytvořeným souborem *.keystore), jinak se mapy v aplikaci nezobrazí. Mapa spolupracuje s aktuální pozicí, která je vyobrazena pulzující modrou tečkou. Samotný MapView (tj.

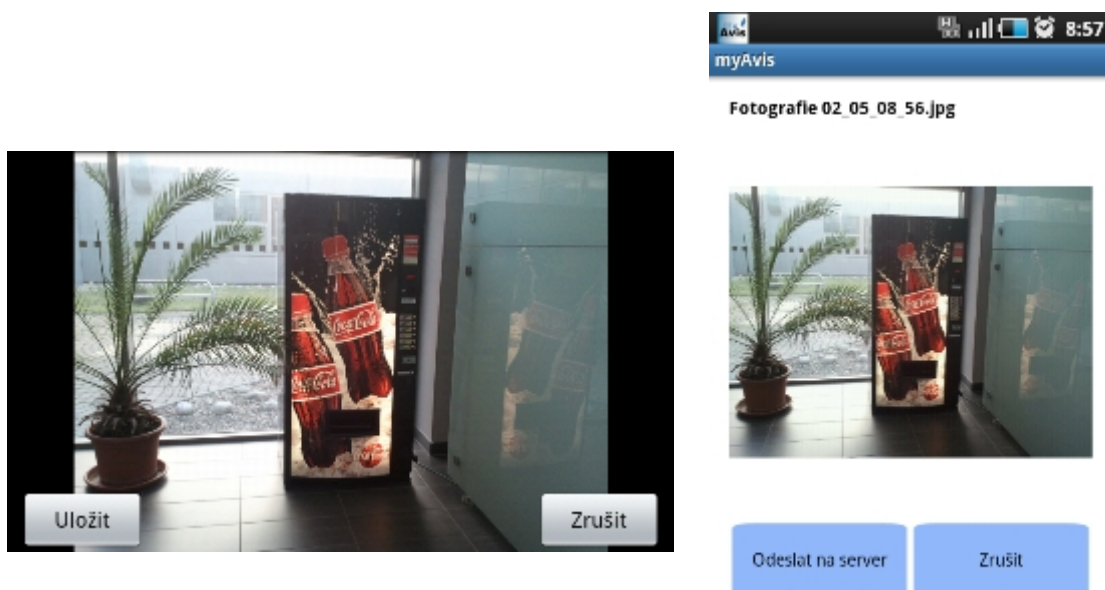
místo, do kterého se načítá mapa) byl vložen do Relative Layoutu, protože jsem k mapě byly přidány i některé podpůrné objekty. Tím prvním je tlačítko, které přepíná mezi zobrazením běžné a satelitní mapy. Dále jsou zde dvě textová pole informující o aktuálních souřadnicích a grafický status přesnosti GPS signálu. Zoomování probíhá na Androidech s podporou technologie multi-touch buď dojíací prstů, popř. se v dolní části objevují tlačítka + a – pro manuální zoomování po jednotlivých krocích.



Obrázek 8: a) Aktuální pozice zobrazená na standardní mapě b) na letecké mapě

Uživatelé budou potřebovat i možnost vyfotit určitou věc, která byla náplní jejich pracovní činnosti, takže bylo dále implementováno fotografování.

Byla využita stávající foto aplikace, kterou má každý Android, vyfocení snímek je jednoznačně pojmenováván stylem `den_měsíc_hodiny_minuty`, např. `15_3_14_06`. Po vyfocení vrací foto aplikace referenci vyfocení snímku, který je zmenšován pro možné potřeby odeslání a fotografie o velikosti v řádech kilobajtů je ukládána do složky `myAvis_photo` nacházející se na paměťové kartě. Není tím narušena konvence focení u Androidu, takže se všechny snímky navíc ukládají v plném rozlišení i do složky `DCIM`, kde je pak můžete uživatel najít.



Obrázek 9: a) Potvrzení vyfoceného snímku b) načtení zmenšené fotky do aplikace

Co se týče dostupných hotových komponent pro Android, které by realizovaly určitou specifickou funkčnost aplikace (např. hotový kalendář, custom list, navigační TabView, apod.), placená řešení se mi podařilo najít pouze u Resco Mobile, to je však k dispozici pouze pro Mono Droid a tedy pro Android nepoužitelné. Ostatní jsou k dispozici ve volně šiřitelných, a mnohdy málo kvalitních, variantách.

2.12 myAvis pro Android – IV. etapa

Poslední pracovní náplní bylo přidání implementace šifrování XML dokumentu, jenž bude umístěn na non-secured FTP serveru, dále pak zjistit možnosti základní navigace v podání již hotových Google maps řešení a v neposlední řadě přidat do menu ikonu, po jejím stisknutí bude spuštěno stahování verze aplikace umístěné na FTP a objeví se instalační nabídka. Bude tedy vynechána kontrola XML a porovnávání verzí z důvodu zjednodušení možného updatu aplikace pro koncové uživatele.

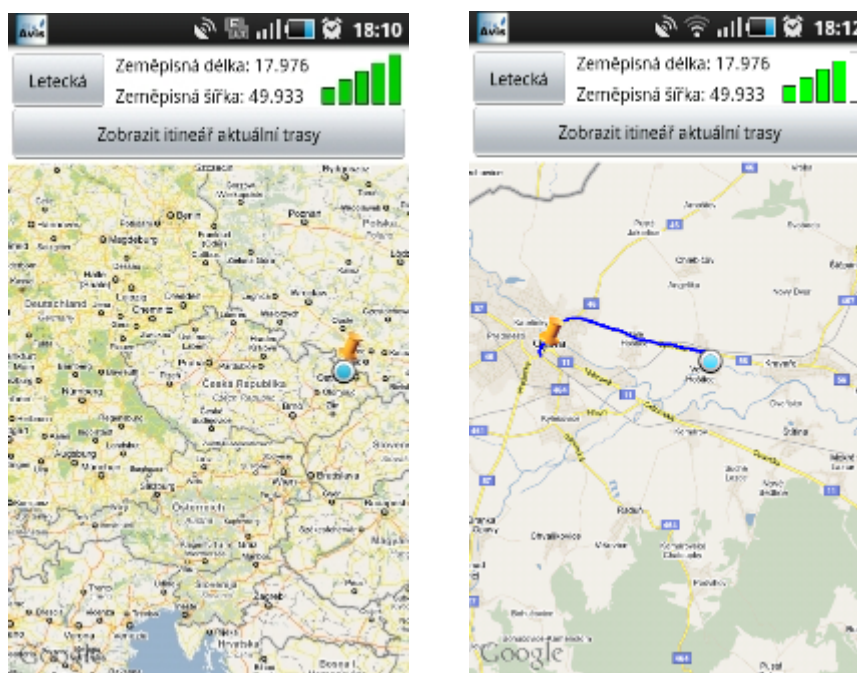
Pro šifrování bylo nejprve potřeba vytvořit SecretKey (tajný klíč) přes příkazový řádek pomocí příkazu keystore, jenž je standardní součástí instalace Javy. Klíč byl generován následovně:

```
keytool -genseckey -alias <alias_klíče> -keystore "<jmeno_souboru.bks>" -provider  
org.bouncycastle.jce.provider.BouncyCastleProvider -providerpath "bcprov-jdk16-146.jar"  
-storetype BKS -storepass <heslo>
```

V průběhu implementace jsem zjistil, že si Android poradí pouze s úložištěm BKS a tedy je potřeba pro správnou funkčnost vytvořit právě BKS klíč. Keytool standardní cestou nechtěl tento klíč vytvořit, protože BKS považoval za neznámý algoritmus. Nakonec pomohlo nalinkování providera Bouncy Castle, což je API třetí strany, které v sobě definici BKS obsahuje. Výsledný klíč je chráněn heslem a uživatel musí pro využití v aplikaci dále znát název klíče a jeho alias.

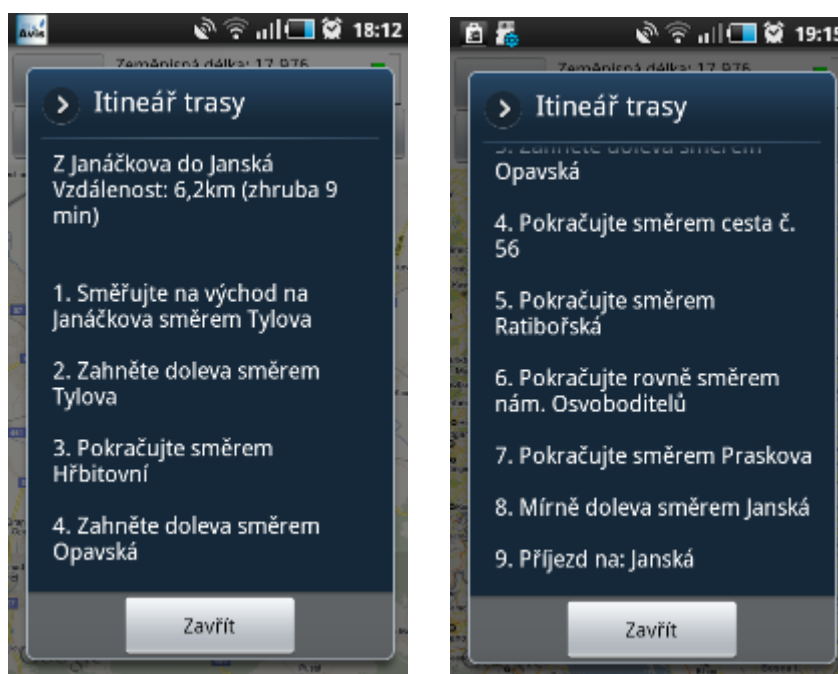
Následně je v programu vytvořena inicializace klíče, načteno heslo ze souboru Values.xml a klíč z APK složky Assets. Dále byla vytvořena třída Encrypter, která se stará o zašifrování a rozšifrování zadaného vstupu, tj. InputStreamu. Využit je při tom CipherOutputStream a CipherInputStream. Výsledný XML dokument na serveru obsahuje verzi souboru na webu, povinnost updatu a URL lokaci pro stažení souboru s aplikací a lokaci pro XML s konfiguračními daty.

Ještě před parsováním XML telefon tento soubor načte a dekoduje, čímž dostává výstup OutputStream. Ten potřebuje předat na vstup parseru, tedy na InputStream. Přes ByteArrayInputStream a ByteArrayOutputStream je docílena konverze OutputStreamu na InputStream tak, aby byla zachována návaznost na původní kód. V případě, že se objeví pokus o zneužití XML souboru a soubor je upraven, proběhne převod, avšak XML parser zachytí chybu, která se zapíše jako výjimka do konzole System.err v LogCatu. Zde se služba updatu zastaví a tedy neproběhne žádný update nové verze, dokud není na server nahrán správný XML soubor.



Obrázek 10: a) Aktuální pozice, cíl a trasa ve velkém měřítku b) Aktuální pozice, cíl a trasa v malém měřítku

Druhým úkolem bylo zjistit, zdali nejsou nějaká hotová řešení základní navigace v aplikaci Google Maps, která již byla do aplikace integrována a je tedy potenciálně vhodná pro další využití. Inspiroval jsem se projektem *j2memaproutepvider*, který lze nalézt na webu code.google.com, a který je distribuován veřejně. Upravil jsem jej a přidal novou funkcionalitu. Základem je získání souboru KML, což je aplikace metajazyka XML pro publikaci a distribuci geografických dat v souřadnicovém systému WGS84. Jako vstupní požadavky pro naplánování trasy slouží souřadnice Latitude a Longitude, které jsou brány z aktuální pozice GPS. Tedy, do té doby, než je zjištěna aktuální poloha uživatele se vizuálně nic neděje. Jakmile je poloha zařizována, zjistí se souřadnice a je sestaveno URL, které je načteno přes metodu `getConnection()`, která se spojí se serverem a vrací `InputStream` s KML souborem. Ten je následně rozparsován SAX Parserem. Do `ArrayListu` se ukládají textové hodnoty itineáře, do pole `Point[] mPoints` se pak ukládají body, které budou v mapě později vykresleny metodou `drawPath`.



Obrázek 11: a), b) Itineář zobrazující počáteční a cílový bod, délku a průjezdní body trasy

V metodě `drawPath` je na místě posledního bodu trasy vykreslen „špendlík“, který vizuálně ukazuje cílovou destinaci. Jeho pozici bylo třeba umístit tak, aby hrot ukazoval přesně na cílový bod trasy. Aktuální pozice je znázorňována modrou poblikávající tečkou, trasa od začátku po cíl je vyobrazena v mapě modrou čarou. Zobrazuje se nad mapovým podkladem, takže může být použita u běžné i letecké mapy. Generování KML souboru je nastaveno při každé změně polohy, datová vytíženost není příliš velká, jelikož každé testovací KML mělo zhruba 80 kB. Po kliknutí na tlačítko `Zobrazit itineář cesty` se zobrazí okno informující o délce cesty a pak o průjezdních bodech. Popisky itineáře jsou stahovány v angličtině (SAX parser české znaky s diakritikou odstraňuje

a vyhazoval chyby), pro českého uživatele bylo vyrobeno překládání stylem nahrazování anglických slov a řetězců za české ekvivalenty. Uživatel je tedy grafikou a textem navigován na cílovou destinaci. Nyní jsou souřadnice cíle nastavovány fixně, v budoucnu mohou být zjišťovány podle aktuálního času, za jak dlouho má uživatel dorazit na určené místo, z databáze.

Posledním pracovním úkonem bylo vytvoření tlačítka v hlavním menu, které po stisknutí začne stahovat update z webu, je-li ve webovém prostoru verze novější. Využit byl stejný kód, jako pro službu, jenž se stará o stahování aplikace, jen z ní bylo vynecháno dotazování uživatele na případnou aktualizaci.

3. Potřebné znalosti získané během studia potřebné pro výkon stáže

Pro potřeby výkonu praxe jsem upotřebil znalosti z předmětů týkajících se programování, uživatelských rozhraní a geoinformačních technologií.

Pro operační systém Android se programuje v jazyku Java, do něž jsem pronikl v předmětech Úvod do Programování (**UPR**) v 1. semestru. Zde byly probrány základy a jednoduché algoritmy. Předmět Programovací Jazyky I (**PJ I**) mi pak přinesl pohled více do hloubky z hlediska znovupoužitelných komponent z pohledu objektového programování a základy paralelismu (tj. využití více vláken).

Pomohly mi i zkušenosti z vývoje aplikace pro mobilní telefony na platformě Java v předmětu Tvorba Aplikací pro Mobilní Zařízení I (**TAMZ I**), v navazujícím předmětu Tvorba Aplikací pro Mobilní Zařízení II (**TAMZ II**) jsem se již dostal v rámci semestrálního projektu k samotné tvorbě aplikace pro operační systém Android v podobě mobilního zpracování britské varianty hry Dáma.

Z pohledu designu jsem využil teoretické znalosti z předmětu Uživatelská rozhraní (**URO**), kde bylo cílem vytvořit desktopové aplikace v jazyku Python a Java s důrazem na uživatelské rozhraní. Cílem tedy bylo vytvořit uživatelské rozhraní, které bude pro uživatele intuitivní, přehledné a funkční. Vnímání designu jsem prohloubil v předmětu Vývoj Internetových Aplikací (**VIA**), kde byl design dvou semestrálních projektů (webová stránka s Ajaxem, PHP a JavaScriptem a webová aplikace/hra v technologii Microsoft Silverlight) tou nejpodstatnější částí.

Při dokončování dokumentu o geoinformačních technologiích jsem čerpal zkušenosti z programů ArcGIS Desktop a ArcPad, které byly hlavní vyučovací náplní předmětu Geoinformační Technologie (**GIT**).

4. Scházející znalosti či dovednosti pro absolvování praxe

Výuka ve škole mi dala rámcovou představu o problémech z oblasti programování a grafiky, přesto je třeba poznamenat, že řada věcí mi při vykonávání praxe scházela. Jednalo se hlavně o specifické dotazy k aplikaci pro Android, které jsem měl při konzultacích se svým vedoucím bakalářské práce. Většinu použitých metod a postupů bylo třeba nastudovat z veřejně dostupných Android API. Nejvíce časově náročná byla implementace Kalendáře a s ním souvisejícího API, které není nikde pořádně popsáno. Konečné řešení tedy spočívalo ve zjišťování hodnot z databázových tabulek při připojení Android zařízení k počítači přes ADB (Android Debug Bridge).

5. Závěr

V průběhu praxe jsem vytvořil komplexní dokument o operačním systému Android, jako příspěvek do IT-Clusteru. Dále jsem zpracoval dokument informující o vlastnostech operačního systému Windows Mobile 6.5/CE z uživatelského i programátorského hlediska, dokument dorazil na VŠB-TUO jako vzorové zadání pro porovnávání mobilních systémů ve vybraných oblastech. Provedl jsem průzkum technologie Adobe FLEX a zhodnotil její potenciální využitelnost pro budoucí projekty spojené s aplikací myAvis. Dále jsem zpracoval krátkou zprávu o vývoji pro Symian, zrecenzoval Motorolu ES400 jako nejnovějšího zástupce odolného smartphonu do terénu s operačním systémem Windows Mobile 6.5. Součástí teoretické části praxe bylo i absolvované školení na téma ArcGIS Mobile ve společnosti VARS Brno a stahování dat na pobočce Sazky v Ostravě.

Praktická část praxe zahrnovala implementaci uživatelského rozhraní aplikace myAvis pro Android, v dalších fázích pak na design navazovaly funkční bloky týkající se implementace Google map, GPS, kalendáře (s možností ukládání dat do interního kalendáře telefonu), fotoaparátu, nahrávání nové verze aplikace pomocí porovnávání verzí XML souborů a šifrování.

Pokud mohu zhodnotit praxi ve firmě Kvados, a.s. ze svého pohledu, hodnotím ji velmi kladně. V průběhu praxe jsem si navykl na plnění úloh, které po mě byly vyžadovány. Vývoj aplikace pro Android je nasměrován správným směrem a k jeho plné funkčnosti je potřeba ještě jeho integrace na interní serverové systémy společnosti. Do budoucna je plánováno dokončování aplikace, testování a její postupné vypuštění na trh, na čemž se jako budoucí zaměstnanec společnosti Kvados, a.s. budu pravděpodobně i nadále podílet.

Seznam Literatury

- [1] BURNETTE, Ed. *Hello, Android : Introducing Google*. 3rd edition. USA : Pragmatic Bookshelf, 2010. 300 s. ISBN 1-934356-56-5.
- [2] MURPHY, Mark L. *Android 2 : Průvodce programováním mobilních aplikací*. Brno : COMPUTER PRESS, 16. 3. 2011. 376 s.
- [3] *Android Developers* [online]. 2011 [cit. 2011-05-02]. The Developer's Guide. Dostupné z WWW: <<http://developer.android.com/guide/index.html>>.
- [4] *Android Developers* [online]. 2011 [cit. 2011-05-02]. Package Index. Dostupné z WWW: <<http://developer.android.com/reference/packages.html>>.
- [5] Gartner Says Android to Challenge Symbian for No. 1 Position by 2014. *Gartner.com* [online]. 10. 9. 2010, Tisková zpráva, [cit. 2011-05-02]. Dostupný z WWW: <<http://www.gartner.com/it/page.jsp?id=1434613>>.
- [6] *Android Developers* [online]. 2011 [cit. 2011-05-02]. What is Android?. Dostupné z WWW: <<http://developer.android.com/guide/basics/what-is-android.html>>.
- [7] *Google User Content* [online]. 2010 [cit. 2011-05-02]. Android 2.2 Compatibility Definition. Dostupné z WWW: <http://static.googleusercontent.com/external_content/untrusted_dlcp/source.android.com/cs/compatibility/2.2/android-2.2-cdd.pdf>.
- [8] Adobe Flex#Adobe Flash Builder and Flex 4. In *Wikipedia : the free encyclopedia* [online]. St. Petersburg (Florida) : Wikipedia Foundation, 2. 11. 2007, last modified on 26. 4. 2011 [cit. 2011-05-02]. Dostupné z WWW: <http://en.wikipedia.org/wiki/Adobe_Flex#Adobe_Flash_Builder_and_Flex_4>.
- [9] *Forum Nokia* [online]. c2011 [cit. 2011-05-02]. Develop. Dostupné z WWW: <<http://www.forum.nokia.com/Develop/>>.
- [10] *GIS Best Practices : Mobile GIS* [online]. USA : ESRI, 2007 [cit. 2011-05-02]. Dostupné z WWW: <<http://www.esri.com/library/bestpractices/mobile-gis.pdf>>.
- [11] *What is Mobile GIS?* [online]. 28. 3. 2011 [cit. 2011-05-02]. MobileGIS. Dostupné z WWW: <http://help.arcgis.com/en/arcgismobile/10.0/help/index.html#/A_quick_overview_of_ArcGIS>.
- [12] SWANSON, Tom. ArcGIS Mobile Deployment Strategies. *Degis* [online]. 2010, Prezentace, [cit. 2011-05-02]. Dostupný z WWW: <http://www.degis.org/images/MobileGIS_DEGIS2010_forDist.pdf>.

Přílohy

Přílohy jsou značeny písmeny A až G a jsou uloženy na CD přiloženém k této Bakalářské práci. Mimo příloh, na něž je v textové části přímo odkazováno, jsou na CD uloženy i ukázky některých zdrojových kódů pro aplikaci myAvis pro Android.

Složka /Bakalářská_práce:

- **Bakalářská práce – Martin Chroust.pdf**
Celý text Bakalářské práce v elektronické podobě

Složka /Přílohy:

- **Příloha A – Android pro IT-Cluster**
Dokument popisující operační systém Android z uživatelského hlediska i z pohledu vývoje
- **Příloha B – Systémové informace o Windows Mobile 6.5 a CE**
Dokument sloužící jako vzor pro porovnávání mobilních systémů ve vybraných oblastech
- **Příloha C – Technologie Adobe FLEX**
Dokument popisující možnosti a případnou vhodnost technologie Adobe FLEX pro možné použití v aplikaci myAvis
- **Příloha D – Vývoj pro Symbian**
Dokument popisující to, co je potřeba pro vývoj pro operační systém Symbian.
- **Příloha E – Recenze Motoroly ES400**
Recenze Motoroly ES400 z hlediska možného využití ve firemních projektech a z pohledu běžného uživatele.
- **Příloha F – absolvování školení ArcGIS Mobile**
Certifikát o absolvování školení ve společnosti VARS Brno.
- **Příloha G – Mobilní Geoinformační Systémy**
Kompletní přehled možností geoinformačních systémů, mobilních GIS a ukázka některých specializovaných zařízení pro sběr dat v terénu
- **Příloha H – Fotogalerie myAvis pro různou hustotu pixelů u displejů**
Fotogalerie displejů různých smartphonů s Androidem za účelem vizuálního porovnání upravených verzí pro jednotlivé hustoty pixelů u displejů

Složka /Ukázkové_zdrojové_kódy:

Složka /Ukázkové_zdrojové_kódy/Java:

- **myAvis.java**
Základní třída programu, ukládání do SharedPreferences, overení přihlášení, vypisování IMEI a IMSI.
- **menu.java**
Třída obsluhující hlavní menu, získávání informací o přesnosti GPS signálu, GPS souřadnice, reverzní geocoding.
- **photo.java**
Třída obsluhující fotografické funkce.
- **tasksAdapter.java**
Třída obsluhující položky v kalendáři, zobrazení dialogů pro výběr data, času a ukládání do interního kalendáře telefonu.
- **downloadFilesTask.java**
Async Task pro stahování souboru z webu.
- **downloadService.java**
Služba běžící na pozadí, která se stará o kontrolu verzí aplikace a případně nabídne alert informující o tom, že je na webu k dispozici nová verze. Integrace šifrování.

Složka /Ukázkové_zdrojové_kódy/XML_layouts:

- **main.xml**
XML soubor pro obrazovku s přihlášením.
- **menu.xml**
XML soubor pro hlavní menu.
- **calendar.xml**
XML soubor pro kalendář.
- **row.xml**
XML soubor definující vzhled každé položky kalendáře.